MODE D'EMPLOI POUR LOGICIEL ICPROG V1.05C
Fonctionne avec les programmateurs :
PIC-01/PIC-02/ATM-01/SER-01/CAR-03/CAR-04

Principe de fonctionnement du buffer:

Les logiciels du programmateur utilisent un buffer, c'est à dire une mémoire intermédiaire entre les fichiers sur disques et les mémoires programmables des composants, tableau hexadécimal visualisé à l'écran.
Pour programmer un composant à partir d'un fichier vous devez d'abord charger le contenu du fichier dans le buffer à l'aide de la commande « Fichier\Ouvrir fichier », puis transférer le contenu du buffer vers le composant avec le menu « Commande\Tout programmer ».
Pour sauvegarder le contenu d'un composant sur disque vous devez d'abord charger le contenu de la mémoire du composant dans le buffer à l'aide du menu « Commande\Tout lire », puis transférer le contenu du buffer sur disque à l'aide du menu « Fichier\Enregistrer sous ».
Lorsque le buffer est chargé à partir d'un fichier sur disque ou à partir de la mémoire d'un composant il est alors possible de modifier son contenu à l'écran en double cliquant sur la valeur à modifier.
Attention, si vous effacer un composant, cela n'effacera pas le contenu du buffer à l'écran pour autant. Vous devez alors refaire une lecture du composant pour voir apparaître les modifications à l'écran.

Le contenu de la fenêtre "Check Sum" (en bas à droite) résulte de l'addition de tous les octets que constitue le programme lu dans le composant. Cela permet par exemple d'identifier un programme par rapport à un autre pour voir s'ils sont identiques. Cette valeur se modifie automatiquement lorsque l'on modifie le programme à l'écran.

Fichier\Ouvrir fichier :

Ouvre un fichier pour le charger dans la fenêtre "Adresse-Programe Code" et "Adresse-Eeprom data". Avant de charger un fichier vous devez avoir sélectionné le type de composant à programmer dans le menu "Configuration\Composant". Vous pouvez lire différent formats de fichiers : .hex (IHX8) pour les fichiers au format Intel 8 bits, .hex (IHX16) pour les fichiers au format Intel 16 bits, .bin pour les fichiers binaires, .mot pour les fichiers au format Motorola-S, .obj pour les fichiers projets et .e2p pour les fichiers au format PonyProg.

Fichier\Enregistrer sous :

Permet de sauvegarder le contenu de la fenêtre courante avec modification possible du nom de fichier et du format de sauvegarde. Si vous sauvegardez avec le format .bin, le contenu du fichier reflétera exactement le contenu de la mémoire du composant. Si vous sauvegardez avec le format .hex le fichier sera codé au format Intel (IHX8).

Fichier\Ouvrir fichier Data :

Normalement, le fichier que vous charger à l'écran dans le buffer se scinde automatiquement dans la partie haute et basse du buffer. Dans certain cas le fichier pour la partie Eeprom Data du microcontrôleur PIC se trouve dans un fichier séparé. Dans ce cas vous utiliserez ce menu pour charger le contenu "Adresse-Eeprom data" du buffer. Vous pouvez lire différent formats de fichiers : .hex (IHX8) pour les fichiers au format Intel 8 bits, .hex (IHX16) pour les fichiers au format Intel 16 bits, .bin pour les fichiers binaires.

Fichier\Fichier récents :

Permet de recharger directement les derniers fichiers utilisés par le programme.

Fichier\Imprimer :

Permet d'imprimer le contenu du buffer en cours.

Fichier\Exit :

Permet de quitter le logiciel. Si des changements ont été apportés le logiciel vous demandera si vous souhaitez enregistrer les modifications.

Edition\Remplir Buffer :

Permet d'initialiser avec la même valeur partout le contenu complet du buffer "Adresse-Programe Code". Cette valeur peut être écrite en décimale, hexadécimale ou ASCII.

Edition\Remplir buffer depuis :

Permet d'initialiser avec la même valeur une portion du buffer "Adresse-Programe Code". Dans le champ « Data » inscrire en décimale, hexadécimal ou ASCII la valeur qui doit être écrite dans le buffer. Dans le Champ « Adresse » inscrire en hexadécimal l'adresse de départ et l'adresse de fin de la portion à modifier.

Edition\Copier Depuis le Buffer et Edition\Coller dans le buffer:
Permet de copier une partie du buffer "Adresse-Programe Code" dans une autre partie du buffer. Sélectionner d'abord avec la souris les codes que vous voulez copier puis appeler le menu « Edition\Copier Depuis le Buffer ». Ensuite, placer la souris à l'endroit à partir duquel vous voulez replacer la portion de code puis appeler le menu « Edition\Coller dna sle Buffer ».

Edition\Remplir Data Buffer :
Permet d'initialiser avec la même valeur partout le contenu complet du buffer "Adresse-Eeprom Data". Cette valeur peut être écrite en décimale, hexadécimale ou ASCII.

Edition\Remplir buffer DATA depuis :
Permet d'initialiser avec la même valeur une portion du buffer "Adresse-Eeprom Data". Dans le champ « Data » inscrire en décimale, hexadécimal ou ASCII la valeur qui doit être écrite dans le buffer. Dans le Champ « Adresse » inscrire en hexadécimal l'adresse de départ et l'adresse de fin de la portion à modifier.

Edition\Copier Depuis le Buffer Data et Edition\Coller buffer Data:
Permet de copier une partie du buffer "Adresse-Eeprom Data" dans une autre partie du buffer. Sélectionner d'abord avec la souris les codes que vous voulez copier puis appeler le menu « Edition\Copier Depuis le Buffer ». Ensuite, placer la souris à l'endroit à partir duquel vous voulez replacer la portion de code puis appeler le menu « Edition\Coller dna sle Buffer ».

Buffer\Active Buffer :
Icprog contient plusieurs buffers que l'on peut activer à l'aide de ce menu.

Buffer\Comparer :
Permet de comparer le contenu de deux buffers. ICprog comparera le buffer actif avec le buffer sélectionné dans le menu. Un message d'erreur apparaîtra en cas de différence entre les deux buffers.

Configuration\Composant\I²C EEprom :
Permet de sélectionner une EEprom du type 24Cxx. S'il s'agit d'une EEprom du type 24LCxx, sélectionner également le composant comme s'il s'agissait d'une 24Cxx. Fonctionne avec les programmateurs PIC-01, ATM-01 et SER-01.

Configuration\Composant\Microchip PIC :
Permet de sélectionner un microcontrôleur PIC du type 12Cxxx, 12Fxxx, 16Cxxx, 16Fxxx, 18Fxxx pour une utilisation avec le programmateur PIC-01. Pour les composant de la série 16C54/55/56/57/58, le mode de programmation est différent et il faut utiliser le programmateur PIC-02.

Différentes options apparaîtront également dans le cadre "Configuration" permettant de modifier les registres de configurations. Pour connaître l'utilisation de ces registres veuillez consulter le datasheet du fabricant concerné. Cependant quelques informations vous sont données ci-dessous pour les microcontrôleurs PIC.

Vous avez le choix entre plusieurs oscillateurs :
- LP : Low power crystal
- XT : Crystal/Resonator
- HS : High speed crystal/Resonator
- RC : Resistor/Capacitor

Cette sélection dépend du type d'oscillateur connecté sur les entrées OSC1/CLKIN et OSC2/CLKOUT lors de l'utilisation du microcontrôleur sur son circuit final après la programmation. Pour les modes XT, LP et HS, un oscillateur à quartz ou un oscillateur TTL/C-MOS est connecté sur les entrées OSC1/CLKIN et OSC2/CLKOUT. Pour le mode RC, un pont RC est connecté sur l'entrée OSC1/CLKIN, (fréquence moins précise).
En général le programme que vous chargez à l'écran sélectionne automatiquement le type d'oscillateur qui sera utilisez lors de l'application.

Validation ou non du WDT :
En validant cette case par une croix, le "Watchdog timer" sera activé. C'est à dire qu'un oscillateur interne indépendant de l'oscillateur externe sera fonctionnel même si le microcontrôleur est en position sommeil.

Validation ou non du PWRT :
En validant cette case par une croix, le "Power-up Timer" sera activé. Le microcontrôleur effectuera à sa mise sous tension un Reset général d'une durée de 72ms, le temps que la tension d'alimentation se stabilise.

Validation ou non du MCLR :
En validant cette case par une croix, le "Memory Clear" sera activé. Il sera possible de faire une remise à zéro externe par la broche “ GP3\MCLR\Vpp ” du microcontrôleur.

Validation ou non du CP :
En validant cette case par une croix, le "Code Protect" sera activé. Le programme intégré dans la mémoire du composant ne sera pas lisible si l’on fait une re-lecture de celui-ci. Cependant le composant reste effaçable pour être reprogrammé si celui -ci contient une mémoire Flash.
Attention si vous cochez cette case, le composant ne pourra pas être vérifié après programmation et un message d’erreur interviendra systématiquement lors de la vérification du composant après programmation.

Identificateur d'un composant :
Dans la case "ID Value", entrer un numéro d’identification qui sera enregistré dans un registre spécial du composant

Configuration\Composant\SPI µC :
Permet de sélectionner un microcontrôleur AVR du type AT90Sxxxxx pour une utilisation avec le programmateur ATM-01.

Configuration\Composant\ « Autres » :
Les classes de composants « 4-Wire Eeprom, IM-Bus Eeprom, Microwire Eeprom, Serial Eeprom, Scenix, SPI Eeprom » sont utilisables avec le programmateur SER-01 uniquement. Voir le fichier « MiseEnOeuvreSER-01.doc » pour de plus amples renseignements.
La classe Flash est supportée par le programmateur ATM-02 uniquement, (pas l’ATM-01).
La classe Serial µC n’est pas supportée par les programmateurs SEEIT.

Configuration\Derniers Composants :
Permet de resélectionner facilement l’un des derniers composants utilisé.

Configuration\Hardware F3 :
Permet de configurer l'interface de programmation entre le logiciel et la carte de programmation.
Programmateur :
JDM programmer pour le PIC-01 / SER-01 / ATM-01 / CAR-01/02/03/04.
TAIT parallel programmer pour le PIC-02
Ports :
COM1 ou COM2 / LPT1 ou LPT2 suivant la connexion de votre programmateur sur votre PC.
Dans tous les cas la LED verte de votre programmateur doit s’allumer lorsque vous effectuez une opération de lecture ou d’écriture. Si ce n’est pas le cas changez de port sélectionné.
Délais I/O :
Ce réglage dépend du PC utilisé, essayez sur 1 ou sur 20 en cas de problème de programmation. Ce réglage n’a pas d’effet sur le port parallèle. (PIC-02).
Interface :
Sélectionner toujours Windows API.
Communication :
Permet d’inverser les signaux envoyés ou reçu sur le port série. En général aucune case n’est cochée.

Pour la configuration exacte en fonction du programmateur utilisé, se référer au fichier « MiseEnOeuvreXXX-XX.doc » se trouvant sur la disquette.

Configuration\Hardware Test:
Permet de tester les différentes broches du port série. En cliquant sur une des cases vous activer un signal haut sur la broche correspondante du port série. Ceci permettant de vérifier le bon fonctionnement d’un programmateur ou de suivre un signal stable sur les différents composants du programmateur.

Configuration\Options\Confirmation :
Permet de faire apparaître une fenêtre de demande de confirmation à chaque fois que l’on doit effectuer une programmation, un effacement, une protection en lecture ou une réécriture.

Configuration\Options\Notification :
Permet de faire apparaître un message lorsque qu’un fichier est chargé à l’écran et qu’il ne contient pas d’ ID-WORD ou de CONFIG-WORD.

Configuration\Options\I²C :
Cette option agit sur la programmation des composants 24Cxx (I²C)
Active MCLR comme VCC :
Permet d'activer la tension de programmation +13V lors de la programmation d'un composant 24Cxx ou d'une carte à bus I²C. Ne cochez pas cette case car vous risquez de griller le composant ou la carte à puce. Cette option n'est à utilisez qu'en connaissance de cause pour faire des tests éventuellement.
Active page write :
Permet d'activer le mode « Page write » permettant de programmer plus rapidement une EEprom 24Cxx. Certaines marques d'EEproms 24Cxx ne supporte pas ce mode de fonctionnement. Dans ce cas décocher cette case mais la programmation sera moins rapide.
Adresse hardware :
Normalement sur 0. Avec le programmateur SER-01, suivant la configuration des cavaliers A0,A1,A2 il est possible de sélectionner une autre valeur de cette adresse pour la programmation du composant 24Cxx.

Configuration\Options\Programmation :
Permet d'effectuer la vérification de la bonne programmation d'un composant après l'avoir entièrement programmé ou durant la programmation du composant. Si aucune case n'est cochée, le composant ne sera pas vérifié et dans ce cas il n'y aura jamais de message d'erreur, que le composant soit bien ou mal programmé.

Configuration\Options\Raccourcis :
Permet de sélectionner 4 raccourcis (CTRL+F1, CTRL+F2, CTRL+F3, CTRL+F4) pour accéder plus facilement à une fonction.

Configuration\Options\Copier&coller :
NC

Configuration\Options\Smartcard :
Permet de configurer la programmation des cartes à puces WaferCard, GoldCard et SilverCardII.
Port :
Sélectionner le port série ou est branchée votre interface Phoenix/smartmouse. Si vous utilisez les programmateurs CAR-03 ou le CAR-04 il faut sélectionner le même port série que celui sélectionné dans le menu « Configuration\Hardware ».
Inverse le Reset (smartmouse) :
Si votre programmateur CAR-03 ou CAR-04 est configuré en mode « Phoenix » ne cochez pas cette case.
Si votre programmateur CAR-03 ou CAR-04 est configuré en mode « Smartmouse » cochez cette case.
Protocol :
Icprog supporte 2 types de Loader, Multimac et Secanix. Sélectionner le mode Multimac pour les cartes à puces.
Composant :
Sélectionner le composant qui se trouve sur votre carte à puce. Pour les PIC16F876 ou les PIC16F877 sélectionner 16F84.
Fréquence :
Sélectionner la fréquence sur laquelle fonctionne votre programmateur : 3,579MHz ou 6,000MHz.
Enable DTR as Vcc :
Cette case est coché normalement.

Configuration\Options\Language :
Permet de sélectionner votre langue favorite pour l'utilisation du logiciel.

Configuration\Options\Shell :
NC

Configuration\Options\Misc :
Priorité:
Permet de définir la priorité du logiciel par rapport aux autres logiciels fonctionnant en multitâches sous Windows. En général utiliser le mode « normal ». Utiliser le mode « haute » pour que ICprog soit prioritaire par rapport aux autres logiciels.
Active Driver NT/2000/XP :
Sous Windows 95/98/ME cette option n'est pas accessible. Sous Windows NT/2000/XP cocher cette case.
Vérifier dans ce cas que le fichier « ICprog.sys » se trouve bien dans le même répertoire que ICprog.exe.
Pour WindowsXP il faut également sélectionner le fichier ICprog.exe et cliquer avec la touche droite de la souris. Lorsque le cadre apparaît, allez dans le menu « Propriétés ». Sélectionner l'onglet « Compatibilité » et cocher la case situé dans le cadre « Mode de compatibilité ». Dans ce même cadre sélectionner « Windows 2000 » dans le menu déroulant.

Active Vcc Control pour JDM :
Ne pas cocher cette case.
Mettre toutes les sorties au niveau haut :
Cette fonction permet de mettre toutes les sorties du port parallèle au niveau haut lorsque le port série est utilisé et de mettre toutes les sorties du port série au niveau haut lorsque le port parallèle est utilisé. Cette fonction sert uniquement lorsque l'on utilise un programmateur spécial ayant à la fois le port série et le port parallèle de connecté sur le PC.

Configuration\Smartcard(Phoenix):
Cocher cette option si vous utilisez les programmateurs CAR-02, CAR-03 ou CAR-04, permettant ainsi d'accéder au modes Smartmouse/Phoenix pour la programmation de l'EEprom 24Cxx sur votre carte à puce. Attention, si vous programmer des EEproms 24Cxx avec le programmateur PIC-01, ATM-01 ou SER-01 il faut décocher cette option, sinon un message d'erreur apparaîtra.

Configuration\Effacer Configuration :
Permet de réinitialiser la configuration et les options du logiciels comme lors de la première utilisation.

Commande\Tout lire :
Permet de lire le contenu d'un composant dans le buffer. La lecture portera aussi bien sur les codes de programme, que sur les données de l'EEprom, que sur le contenu des registres de configurations.
Si le composant est protégé en lecture, le buffer n'indiquera pas les bonnes valeurs à l'écran et affichera des 0. Dans ce cas s'il s'agit d'une mémoire Flash le composant pourra être effacé et reprogrammé normalement.
Si vous voulez sauvegarder le contenu d'un composant sur disque, vous devez d'abord utiliser cette fonction, puis utiliser la fonction "Enregistrer Sous".
Si vous voulez faire une copie d'un composant, vous devez d'abord utiliser cette fonction pour charger le buffer, changer le composant, puis utiliser la fonction "Command\Tout Programmer".

Commande\Tout Programmer :
Permet de programmer le contenu d'un composant à partir du buffer.
La programmation portera aussi bien sur les codes de programme, que sur les données de l'EEprom, que sur le contenu des registres de configurations. Dans ce cas assurez vous bien que les registres de configuration sont bien positionnés avant de lancer cette fonction.

Commande\Programmer Configuration :
Permet de programmer uniquement les registres de configuration du composant.

Commande\Tout Effacer :
Permet d'effacer le contenu du composant.
Tous les bits dans ce cas sont remis au niveau logique 1, (hFF). Cette fonction n'est pas toujours disponible car certains composants ne sont programmables qu'une fois.

Commande\Test de virginité :
Permet de faire un test de virginité pour vérifier que le composant est vide.
Si le composant est vierge ou effacé tous les bits de la mémoire seront au niveau logique 1 (FF...). Cette fonction est à utiliser avant toute programmation car il n'est pas possible de programmer un composant correctement si celui-ci n'est pas vierge ou n'a pas été effacé préalablement. En effet une programmation met certains bits au niveau logique 0, l'inverse n'étant possible que si le composant peut être effacé à l'aide de la fonction "Command\Tout Effacer".

Commande\Vérifier :
Vérifie le contenu du composant avec celui du buffer. Un message d'erreur apparaît en cas de différence.

Commande\Assistant Smartcard :
Assistant permettant la programmation d'une carte à puce, pas à pas, suivant la procédure suivante.
Avec le CAR-03 et le CAR-04, lorsque le logiciel demande de déconnecter le programmateur de PIC et de connecter le programmateur Phoenix, il vous suffit de déplacer le cavalier se trouvant sur le programmateur, de la position JDMprog à la position Phoenix et vice versa.

Procédure :
Vérifier que le cavalier situé sur le CAR-04 est bien sur la position « JDMprog ».
Insérer la carte à puce, contact vers le bas, dans le lecteur puis mettre le programmateur CAR-04 sous tension.
Cliquer sur « Continuer ». Le Loader permettant l'accès à l'EEprom se charge automatiquement dans le PIC16F84.
Déplacer le cavalier situé sur le programmateur CAR-04, sur la position « Phoenix ».

Cliquer sur « continuer ». Sélectionner le fichier pour l'EEprom 24LC16B. Cliquer sur Ouvrir.
Cliquer sur « Continuer ». Le programme se charge dans l'EEprom 24LC16B.
Remettre le cavalier sur la position « JDMprog ».
Cliquer sur « continuer ». Sélectionner le fichier pour le PIC16F84. Cliquer sur Ouvrir.
Cliquer sur « Continuer ». Le programme se charge dans le PIC16F84.

Outil\CRD upload :
Permet de convertir le programme afficher dans le buffer en code assembleur.

Voir\Assembleur :
Permet de convertir le programme afficher dans le buffer en code assembleur.

Voir\Voir en Hex :
Permet de revenir à l'affichage en hexadécimal du buffer.

Voir\Emplacement composant :
Permet de voir le positionnement du composant sélectionné sur le support.
Ce menu n'est pas utile avec les programmateurs SEEIT car tous les supports tulipes sont existant sur le programmateur et il n'est dont pas nécessaire d'implanter de façon particulière ou avec un adaptateur un composant sur les supports.

Voir\Information composant :
Permet d'obtenir quelques renseignements sur le composant sélectionné :
Nom, type, taille de la mémoire, architecture mémoire, nombre de pattes.

Voir\Rafraichir :
Permet de réinitialiser l'écran. A utiliser l'été uniquement.

LISTE DES PRODUITS PROPOSES PAR SEEIT:

Informations complémentaires sur le site http://www.seeit.fr

Programmateur de cartes PCMCIA :
CAS-02 : programmateur de cartes PCMCIA (cartes CAM)

Programmateur de cartes à puces :
CAR-04 : Multi-Programmateur de cartes à puce compatible
Phoenix / Smartmouse / I2C Bus / PIC-JDMprog / AVR-SPIprog sur port série.
CAR-05 : Programmateur de cartes à puces PIC/AVR/I2C/T2G sur port parallèle.
CAR-06 : Programmateur universel de cartes à puce PIC et Atmel sur port USB
MASTERA-4 : Programmateur de cartes à puces multimodes sur port série et
duplicateur autonome.
MULTIPROG-4 : Programmateur de cartes à puces sur port USB avec émulation Phoenix 3,58 et 6,00 MHz.

Programmateurs de microcontrôleurs :
ATM-01 : Programmateur de microcontrôleurs AVR de Atmel et de 24Cxx.
PIC-01 : Programmateur de microcontrôleurs PIC de Microchip et de 24Cxx, +110 composants).
PIC-02 : Programmateur de microcontrôleurs PIC16C54/16C5x
PIC-03 : Programmateur de microcontrôleurs PIC17C42/17C4x
PSTART : Programmateur universel de microcontrôleurs PIC de Microchip.

Programmateurs d'E(E)proms et Flash Eproms :
SER-01 : Programmateur d'EEproms séries, support DIP8.
EPR-01 : Programmateur d'Eproms et d'EEproms parallèles, 512 Kb, support DIP28
EPR-02 : Programmateur d'Eproms et d'Eeproms parallèles, 8 Mb, support DIP32.
EPR-03 : Programmateur de BIOS et mémoires FLASHs, 512Kb à 4Mb, +200 composants, support DIP32 et PLCC32.
LPC-32 : Programmateur universel d'E(E)proms et Flash Eproms 8 Mb sur PC, +200 composants, support DIP32.
LEAPER-3C : Programmateur d'EPROMs et mémoires FLASHs 8 Mb autonome ou sur PC, 2 x DIP32, port USB.

Programmateurs universels d'E(E)proms, Flash Eproms, PLD, Proms, Microcontrôleurs (produits professionnels):

SUPERPRO-Z : Programmateur universel, +1600 composants, support DIP40 sur port parallèle.
CHIPMAX : Programmateur universel, +1600 composants, support DIP40 sur port parallèle.
SUPERPRO-280U : Programmateur universel, +3000 composants, support DIP48 sur port USB.
SUPERPRO-580U : Programmateur universel, +4500, composants, support DIP48 sur port USB.
SUPERPRO-3000U : Programmateur universel, +6000 composants, support DIP48 sur port USB.
LEAPER-48 : Programmateur universel, +2000 composants, support DIP48 sur port USB.
TOPMAX : Programmateur universel, +4600 composants, support DIP48 sur port parallèle.

Multicopieurs universels de productions :
MULTIMAX-1G : Multi-Programmateur universel, 1 support DIP48, autonome ou sur port USB.
MULTIMAX-8G : Multi-Programmateur universel, 8 supports DIP48, autonome ou sur port USB.
MULTIMAX-8G : Multi-Programmateur universel, 16 supports DIP48, autonome ou sur port USB.
TOPMAX-8GANG : Programmateur 8 supports ZIF interchangeables sur port parallèle.

Adaptateurs pour programmateurs :
ADA-SOIC16-150 : Adaptateur universel pour les EEPROMs CMS en 8, 14, 16 pattes (largeur 150mil).
ADA-SOIC18-300 : Adaptateur universel pour microcontrôleur PIC ou autres en boîtiers SO18 (largeur 300mil).
ADA-SOIC28-300 : Adaptateur universel pour microcontrôleur PIC ou autres en boîtiers SO28 (largeur 300mil).
ADA-PLCC32 : Adaptateur universel DIP28-DIP32/PLCC32 pour E(E)proms et Flash Eproms.
ADA-PLCC44 : Adaptateur universel DIP40/PLCC44 pour E(E)proms et microcontrôleurs.
ADA-TSOP32-14 : Adaptateur universel DIP32/TSOP32 pour Flash EPROMs (largeur=14mm)
ADA-TSOP32-20 : Adaptateur universel DIP32/TSOP32 pour Flash EPROMs (largeur=20mm)
ADA-TSOP48-20 : Adaptateur universel DIP48/TSOP40-TSOP48 pour Flash EPROMs (largeur=20mm)
PA44-48U : Adaptateur universel DIP48/PLCC44 pour TOPMAX.
PA44QF44D : Adaptateur universel DIP44/QFP44 pour TOPMAX.
PA44SO44D : Adaptateur universel DIP44/SOP44 pour TOPMAX.
TM-8G-DP : Adaptateur GANG 8xDIP32 pour programmer par 8 les E(E)proms et Flash sur le TOPMAX.

Emulateur d'EPROMS, EEPROMS et mémoires FLASHS :
EEROM-8U : Emulateur d'EPROMS, EEPROMS, et mémoires FLASHS, 8Mb, 45ns, sur port USB.

Effaceurs d'Eproms :
LER-121A : Effaceur d'Eproms en coffret avec minuterie pour 12 composants.
LER-123A : Effaceur d'Eproms en coffret avec minuterie pour 64 composants.

Testeurs de câbles :
PC-CABLE-PRO : Testeur de cordons informatique pour PC.
LANTEST : Testeur de câbles réseaux avec afficheur à LEDs.
LANTEST-PRO : Testeur de câbles réseaux RJ45.